UDC 004.31

DESIGN MODELS OF BIT-STREAM ONLINE-COMPUTERS FOR SENSOR COMPONENTS

Larchenko L. V. – PhD, Associate Professor, Associate Professor of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Parkhomenko A. V. – PhD, Associate Professor, Associate Professor of the Department of Software Tools, National University “Zaporizhzhia Polytechnic”, Zaporizhzhia, Ukraine.

Larchenko B. D. – PhD, Assistant of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Korniienko V. R. – Postgraduate student of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

ABSTRACT

Context. Currently, distributed real-time control systems need the creation of devices that perform online computing operations close to the sensor. The proposed online-computers of elementary mathematical functions can be used as components for the functional conversion of signals in the form of pulse streams received from measuring sensors with frequency output.

Objective. The objective of the study is the development of mathematical, architectural and automata models for the design of bit-stream online-computers of elementary mathematical functions in order to create a unified approach to their design, due to which the accuracy of calculating functions can be increased, functional capabilities expanded, hardware costs reduced, and design efficiency increased.

Method. Mathematical models of devices were developed using the method of forming increments of ascending step functions based on inverse functions with minimization of calculation error. Automata models of online-computers based on Moore’s Finite State Machine have been developed, the graph diagrams of which made it possible to ensure the clarity of function implementation algorithms, to increase visibility and invariance of implementation in formal languages of programming and hardware description.

Results. The paper presents the results of research, development and practical approbation of design models of bit-stream online-computers of power functions and root extraction function. A generalized architecture of an online-computer was proposed.

Conclusions. The considered functional online-computers are effective from the point of view of calculation accuracy, simplicity of technical implementation, and universality of the architecture.

KEYWORDS: functional conversion, bit-stream data, bit-stream computing, mathematical model, finite state machine, FPGA, SoC.

ABBREVIATIONS

ASIC is an Application-Specific Integrated Circuit;

CAD is a Computer Aided Design;

CMOS is a Complementary Metal Oxide Semiconductor;

CORDIC is a Coordinate Rotation Digital Computer;

FPGA is a Field Programmable Gate Arrays;

FSM is a Finite State Machine;

HDL is a Hardware Description Language;

PWM is a Pulse-Width Modulation;

VHDL is a VHSIC Hardware Description Language;

VHSIC is a Very High-Speed Integrated Circuits;

VSF is a Virtually Scaling Free.

NOMENCLATURE

$|\delta_{\max}|$ is an absolute error limit value of continuous ascending functions reproduction;

$1-|\delta_{\max}|$ is a level of the function approximation node;

Δ_{y-1} is a difference obtained when comparing the increments current values of the functions;

$\Psi(y-|\delta_{\max}|)$ is an inverse function of $f(x)$;

$f(x^{*})$ is a continuous function;

$f(x)$ is an approximating function;

$f(x_y)$ is a value of the function at the point x_y;

$f(x_y-1)$ is a value of the function at the point (x_y-1);

m is a numerator of the fractional exponent of the power function, positive natural number;

n is a denominator of the fractional exponent of the power function, positive natural number;

x is an input bit data stream;

x_y is a value of selected bit from the input bit-stream x (sample);

y is an output bit data stream;

y_k is a value corresponding to the node of the function approximation.

INTRODUCTION

During the development of real-time systems focused on sensor systems, the Internet of Things and distributed control systems an important task is the alignment of sensors with digital systems for collecting and processing information.

Currently, an integral part of new basic elements development needed for the mentioned systems’ creation is the design of specialized hardware functional converters and online-computers, which utilize data in bit (pulse) stream form as an information signal, which allows simpler implementation of operations compared to other types of coding [1–4].

The object of the study is the process of designing bit-stream online-computers of elementary mathematical functions.

The subjects of the study are design models of bit-stream online-computers.

The objective of the study is the development of mathematical, architectural and automata design models

DOI 10.15588/1607-3274-2024-1-6

of bit-stream online-computers of elementary mathematical functions in order to create a unified approach to their design, due to which the accuracy of calculating functions can be increased, functional capabilities expanded, hardware costs reduced, and design efficiency increased.

To achieve the objective, it was necessary to fulfill the development of mathematical models of bit-stream online-computers based on method analysis for generating increments of increasing step functions with minimization of calculation error; the creation of generalized architecture model of online-computer; development of automata models of the device based on FSM; the experimental research of the designed hardware model of a bit-stream power function computer in the hardware description language VHDL using an automata template and implementation in a Xilinx FPGA.

The paper is organized as follows. Sections 1 and 2 provide a description of problem statement and current state of the art. Section 3 discusses the method of forming increments of ascending step functions and mathematical models of online-computers obtained on its basis. A generalized online-computer architecture is presented, which allows building of pipeline architectures of online-computers of specific functions. An approach to the design of online computers based on FSMs was considered, which made it possible to develop a state diagram of the control automaton of online computers and an algorithm flowchart of the operating automaton for the implementation of functions. Sections 4, 5 presents the results of an experimental study of hardware implementation of online-computers for power functions and root extraction function, which confirm the results of theoretical developments. Section 6 provides short discussion about main achievements of this work.

1 PROBLEM STATEMENT

In bit-stream coding, data is presented as streams of unit amplitude pulses. In bit-stream data the informative parameter is fixed value of pulses of arbitrary duration for a time interval. The bit-stream form of signals allows the transmission and processing of information in ways that make it possible to sequentially process single bits of the stream when they are fed to the input of the device [5].

At the same time, the bit-stream form of signals, while maintaining immunity to interference, does not suffer from information redundancy and allows for high-speed operation of devices. Data processing involves both the conversion of the form of information presentation and the linearization of the sensor signal using various elementary mathematical functions [6, 7].

Moreover, bit-stream online-computers provide for the implementation of the stream method of online calculations with simultaneous parallel-serial execution of conversions over single bits of the stream according to the required function. At the same time, sequential calculation of function values is performed for adjacent values of the argument. Stream methods of processing information signals are characterized by the possibility of implementing functional conversion through the use of methods of forming increments, as well as the implementation of sequential processing of streams in the process of receiving single bits [7, 8].

Bit stream online-computers of power functions and root extraction functions can be used in distributed control systems as components for functional conversion of bit streams obtained from measurement sensors of physical quantities with frequency output. The frequency output signal can be a stream of pulses or a signal with PWM [9, 10].

The question of analyzing the methods and means of reproducing functions using converters and online-computers arises in connection with their wide selection and the variety of solved tasks, their specifics and technical requirements. Each selected type of means corresponds to its own calculation method, and rational means of its implementation must be chosen for any method.

In the creation of nonlinear converters and online-computers of bit-stream data which operate in real time for the reproduction of elementary functions, accurate methods of calculating approximating functions based on differential equations or algebraic equations (inverse functions) are used. Thus, the design of specialized hardware functional converters and computers performing bit-stream data conversion in real time is relevant.

2 REVIEW OF THE LITERATURE

In works [11, 12] the need to develop online-computers for processing data which are located close to the sensor, for use in distributed sensor systems is substantiated, stream processing algorithms are shown, which provide functional conversions of signals presented in the form of a bit stream (streams of single pulses, streams of PWM signals) and binary codes at the same time. The proposed principles of stream processing algorithm organization are considered as a continuous process of result formation.

Work [13] presents an approach to the design and hardware implementation of a converter that approximates a function for an argument that is a PWM signal. In this case, an approximating function in the form of dividing polynomials is used. In work [14], the proposed device is focused on the processing of measurement results, which is represented by a pulse stream, in which the functional conversion of information is carried out in the tracking mode.

In work [15] the structure of the converter of the time-frequency signals' parameters into a digital code based on a radial neural network is proposed. A decomposition of the converter into two components is proposed, in which the second component is a radial base network.

One of the relevant directions is the further development of specialized control devices for aerospace systems, in which the most important requirements for built-in online-computers and converters are low energy consumption, the ability to perform fast rough calculations, and low memory consumption. The CORDIC algorithm is used as a mathematical apparatus for such systems.

DOI 10.15588/1607-3274-2024-1-6

Works [16, 17] present a conceptual design approach, and realization of CORDIC architectures. The CORDIC is reconfigurable and can function in either of two modes: for hyperbolic or for circular trajectories in rotation and vectoring. In work [18] the principle of designing and implementing FPGA various trigonometric and logarithmic functions using CORDIC algorithm are presented.

A new efficient modified CORDIC algorithm is proposed. Work [19] explores approximation in CORDIC architectures for CMOS ASIC implementation.

A new efficient modified CORDIC algorithm is proposed in work [20], which combines the conventional CORDIC algorithm with the VSF CORDIC algorithm modelled. In work [21] divider architectures are proposed based on the Newton Raphson division using the reciprocal operation.

The design process can be carried out using tools of automated design systems based on hardware description languages for further synthesis and implementation into the FPGA platform, which ensures configuration flexibility, high speed, and technological reliability [22].

3 MATERIALS AND METHODS

Let's consider a method for forming increments of ascending step functions.

When designing bit-stream online-computers of elementary mathematical functions, a method of forming increments of ascending step functions during functional processing of bit streams corresponding to certain integer values of the argument is proposed.

In bit-stream functional online-computers, the reproduced function changes its value at discrete points and by a discrete value. At the same time, the process of calculating a continuous function involves calculating a lattice function that approximates a continuous one. The main criterion that determines the effectiveness of approximation for computing problems is the approximation error. The search for approximating expressions implemented in hardware is carried out taking into account additional restrictions of the expression type determined by the element base used.

The method of forming increments based on inverse functions ensures the fulfilment of requirements for functional online-computers, which include accuracy and calculation time, simplicity of technical implementation, and universality of using the architecture from the point of view of using the device to perform other mathematical operations. As a result, the method allows applying a unified approach to the design of online-computers. When synthesizing functional online-computers, the absolute error of the calculation must be rational, that is, to provide for all integer values of the argument the limiting value of the absolute error of the calculation ±0.5 of the least significant bit of the argument.

The input x and output y information signals of the considered online-computers are two periodic bit sequences. At the same time, the periodicity of the presentation of bits of the input sequence x is determined by the method of quantization of the reproduced function, and the periodicity of the stream of output bits of the sequence y is determined by the device's functioning algorithm. At the same time, uniform quantization of the argument with integer values is ensured.

The implementation of the approximating function in the online-computer includes certain stages.

At the first stage, an approximating function is selected for a given continuous one and absolute calculation error value at integer points of the argument is determined.

At the second stage, a mathematical model of the online-computer is developed using the considered stepwise approximation method based on inverse functions.

The model establishes a functional relationship in the form of a system of difference inequalities between the bit numbers of the output data stream $y = 1, 2, 3, ..., k$ and their corresponding values $x_y = x_1, x_2, x_3, ..., x_i$ of the input data stream x.

At the third stage, the architecture of the online-computer is created.

At the fourth stage, the hardware implementation of the online-computer is performed using CAD tools, which includes the creation of an automata model of the device based on a state machine, the implementation of the automata model in HDL code for the verification of the behavioral model and further synthesis and implementation in the FPGA.

A continuous function $y^* = f(x^*)$, the restrictions of which are the conditions of $x^*, y^* \geq 0,\ y^* \leq x^*,\ dy^*/dx^* > 0$, which has an inverse $x^* = \psi(y^*)$, and is reproduceable at the output of a hardware bit-stream online-computer by a function that approximates a continuous one:

$$y = [f(x) + |\delta_{\max}|]. \tag{1}$$

The value of absolute error limit value of continuous ascending functions reproduction $\delta_{\max}$ is in the range $0.5 \leq |\delta_{\max}| < 1$.

Continuous and its approximating step functions are shown in Fig. 1.

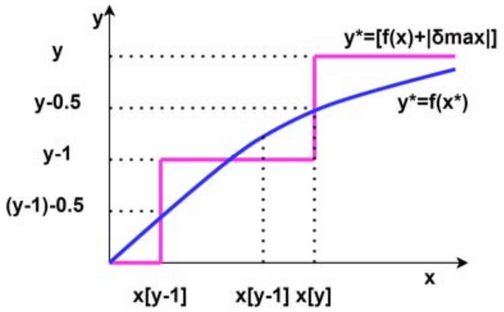

Figure 1 – Approximating step function

For any level $1-|\delta_{\max}|$, $y = 1, 2, ..., k$ it is possible to specify a pair of integer values $x_y - 1$ and x_y for which a system of inequalities (2) takes place:

$$\begin{cases} f(x_y - 1) < y - |\delta_{\max}|, \\ f(x_y) \ge y - |\delta_{\max}|. \end{cases} \quad (2)$$

The values $x_y = x_1, x_2, x_3, ..., x_i$ correspond to the starting moments of the formation of each $y = 1, 2, 3, ..., k$ step of the approximating function.

The process of reproducing the function (1) with the bit-stream form of the argument can be performed on the basis of sampling certain bit values x_y from the input stream x_y, the numbers of which are determined using the inequality (3):

$$\Psi(y - |\delta_{\max}|) \le x_y < \Psi(y - |\delta_{\max}|) + 1. \quad (3)$$

The values x_y of selected bits from the input bit-stream x (sample) are obtained by substituting integer values in inequality (3).

The search for approximating expressions implemented in hardware is carried out taking into account additional restrictions on the type of expression determined by the element base used.

Inequality (3) is the general term x_y formula of the numerical sequence of bits formula of the numerical sequence of bits $x_1, x_2, x_3, ..., x_i$, which are selected from the input bit stream x and correspond to the approximation nodes of the step function y.

Integer sample which are selected from the input bit stream x and correspond to the approximation nodes of the step function y. Integer sample values x_y can be found by sequentially substituting $y = 1, 2, 3, ..., k$ into inequality (3) with the minimum absolute error of calculations $|\delta_{\max}|$=0.5 sample values.

The considered method facilitates the reproduction of function in real time during the arrival of the bit stream x at the online-computer input and is rational from the point of view of the functions' reproduction time, which is always less than the length of the bit stream x and the accuracy of functions' reproduction for integer values of the argument with the calculation error of ±0,5 units of the argument's least significant bit.

With the minimum absolute error of calculations, the values of samples x_y are defined according (4), (5):

$$\Psi(y - 0.5) \le x_y < \Psi(y - 0.5) + 1, \quad (4)$$

$$x_y = [\Psi(y - 0.5) + 1]. \quad (5)$$

The considered method of approximation was used in the development of mathematical models of hardware bit-stream computers which reproduce power and irrational functions.

As an example, consider the mathematical models of the online-computer of power function and the computer of root extraction function.

The power approximating function that reproduces the continuous function at the computer's output has the form (6):

$$y = [x^{\frac{m}{n}} + 0.5]. \quad (6)$$

Sample values x_y can be determined on the basis of inequality (4) and the absolute calculation error of $|\delta_{\max}|$=0.5. Therefore, the inequality implemented in the online-computer has the form (7):

$$2^n x_y^m \ge (2y_k - 1)^n. \quad (7)$$

As a result, the mathematical model of the online-computer of power functions is a system of difference inequalities and has the form (8):

$$\begin{gathered} 2^n x_1^m \ge (2y_1 - 1)^n, \\ 2^n (x_2^m - x_1^m) + \Delta_1 \ge (2y_2 - 1)^n - (2y_1 - 1)^n, \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ 2^n (x_y^m - x_{y-1}^m) + \Delta_{y-1} \ge (2y_k - 1)^n - (2y_{k-1} - 1)^n. \end{gathered} \quad (8)$$

It is given that Δ_{y-1} is the difference obtained when comparing the increments current values of the functions $2^n x_y^m$ and $(2y_k - 1)^n$ between two adjacent approximation nodes of the reproducible function at the previous comparison step; Δ_1 is the difference obtained at the first step of comparison, when performing the first inequality of system (8); values $y_1 \le y \le y_k$ and $1 \le y_k \le k$.

In the system of inequalities (8) Δ_{y-1} defined as:

$$\Delta_{y-1} = 2^n (x_y^m - x_{y-1}^m) + \Delta_{y-2} - (2y_k - 1)^n + (2y_{k-1} - 1)^n. \quad (9)$$

The online-computer of power functions can operate in two modes.

If the fractional power exponents $m<n$, then the device works in the mode of sampling of bits x_y from the bit stream data and forms an output stream y_k. If the fractional power exponents $m>n$, then the online-computer works in the bit sequence series generation mode.

It should be noted that if the online-computer works in the bit sampling mode, then when each inequality (8) is executed, the output bit y_k will be formed at the output of the computer. If the online-computer works in the bit sequence generation mode, a series of bit sequences will be generated at the output of the device when each inequality (8) is executed.

The mode of sampling of bits x_y from the bit stream data in the online-computer of power functions is considered in work [23]. This work presents a mathematical model of an online-computer of a power function, in which the exponent of the power is $m>n$.

The power function online-computer reproduces at the output the approximating function, which has the form (10):

$$y = [x^{\frac{3}{2}} + 0.5]. \quad (10)$$

Based on (7), the inequality realized in the device has the form (11):

$$2^2 x_y^3 \ge (2y - 1)^2. \quad (11)$$

DOI 10.15588/1607-3274-2024-1-6

The difference inequalities of the mathematical model of the calculator on the basis of equations (12), (13):

$$2^2 x_y^3 = 2^2 x_y^3 - 2^2 x_{y-1}^3 + 2^2 x_{y-1}^3 \qquad (12)$$

$$(2y_k - 1)^2 = (2y_k - 1)^2 - (2y_{k-1} - 1)^2 + (2y_{k-1} - 1)^2. \qquad (13)$$

The mathematical model of the power function online-computer is described as a system of inequalities (14):

$$\begin{gathered} 2^2 x_1^3 \ge (2y_1 - 1)^2, \\ 2^2 (x_2^3 - x_1^3) + \Delta_1 \ge (2y_2 - 1)^2 - (2y_1 - 1)^2, \\ \dots\dots\dots\dots\dots\dots, \\ 2^2 (x_y^3 - x_{y-1}^3) + \Delta_{y-1} \ge (2y_k - 1)^n - (2y_{k-1} - 1)^n. \end{gathered} \qquad (14)$$

Here Δ_{y-1} is defined as (15):

$$\Delta_{y-1} = 2^2 (x_y^3 - x_{y-1}^3) + \Delta_{y-2} - (2y_k - 1)^2 + (2y_{k-1} - 1)^2 \qquad (15)$$

In the system of inequalities (14), when each inequality of the system is implemented, a series of bit sequences will be formed.

If the numerator of the fractional exponent of the power function m=1, then from the power function you can go to the function of extracting the root.

The online-computer of the square root function should reproduce the approximating function at the output (16):

$$y = [\sqrt{x} + 0.5] \qquad (16)$$

Function (16) has an inverse function, so using (4) we obtain the inequality implemented in the device:

$$2^2 x_y \ge (2y_k - 1)^2 \qquad (17)$$

The x_y values selected from the input bit stream and supplied to the output of the device are determined by the formula:

$$x_y = [(y - 0.5)^2] + 1. \qquad (18)$$

On the basis of (17), a mathematical model of the bit-stream computer of the square root extraction function is obtained, which is described by a system of inequalities (19):

$$\begin{gathered} 2^2 x_1 \ge (2y_1 - 1)^2 \\ 2^2 (x_2 - x_1) + \Delta_1 \ge (2y_2 - 1)^2 - (2y_1 - 1)^n, \\ \dots\dots\dots\dots\dots\dots \\ 2^2 (x_y - x_{y-1}) + \Delta_{y-1} \ge (2y_k - 1)^2 - (2y_{k-1} - 1)^2 \end{gathered} \qquad (19)$$

When the bit x_y selected from the input stream x is received at the input of the device, the output bit y_k will be formed at its output when each inequality (19) is fulfilled.

Let's consider a generalized architecture of the bit-stream online-computer.

The work proposes a unified approach to the creation of architectural models of bit-stream computers, while taking into account the principles of the computing process' organization based on the analysis of the obtained mathematical models, which allows for the synthesis of architectural solutions. Performing computational procedures in a single way allows the use of a narrow nomenclature of components in such models, which makes it possible to use unified blocks. In Fig. 2 a generalized architecture of a bit-stream computer is given.

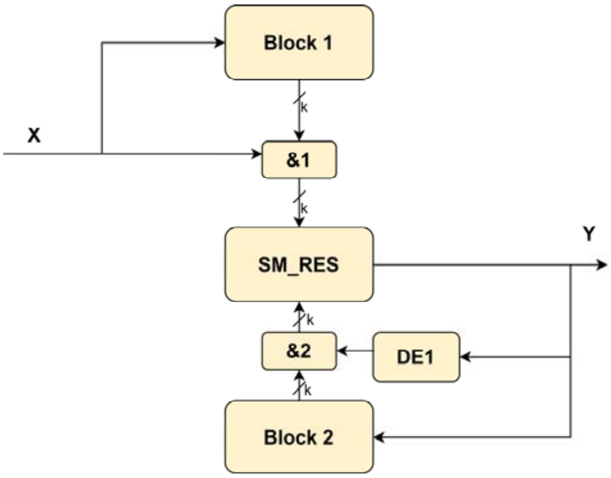

Figure 2 – Generalized architecture of the online-computer

The main computing core of the online-computers is the SM_RES parallel adder with feedback, which is used as a parallel code comparison component and compares the increments of two simultaneously reproduced ascending step functions. The increments of the argument function are given to SM_RES by the bits of the input bit-stream x in direct parallel binary code, and the increments of the function representing the levels of the reproducible step function y by its output bits in the additional code. The SM_RES overflow bits correspond to the beginning of each successive step formation of the approximating function of the device.

The architecture of the bit-stream online-computer includes: blocks Block1 and Block2, which are pipeline architectures built on parallel adders and form increments of the lattice functions of the left and right parts of the inequalities of mathematical models of online-computers.

Let's consider the automata models of bit-stream online-computers.

The proposed hardware implementation approach of the Moore's FSM based online-computers allowed to create a general graph of transitions for the online-computers' control unit and flowchart of the operating unit algorithms for the realisation of considered mathematical functions and to develop corresponding HDL models, for the implementation of reproducible functions' algorithms for the proposed bit-stream computers with CAD/PLUS tools.

DOI 10.15588/1607-3274-2024-1-6

A bit-stream computer represented by a composition of control firmware and operating unit.

The operating unit has computational states in which pipeline calculations are performed, initiated by the signals of the control unit.

At the same time, the control unit determines the order of following the control signals and performing the sequence of microoperations based on the algorithm flowchart of the and the set of warning signals generated by the operating unit.

The graph was obtained as a result of marking the algorithm flowchart of the computer's arithmetic block (Fig. 3) and made it possible to increase the clarity and visibility of the device's computing states control.

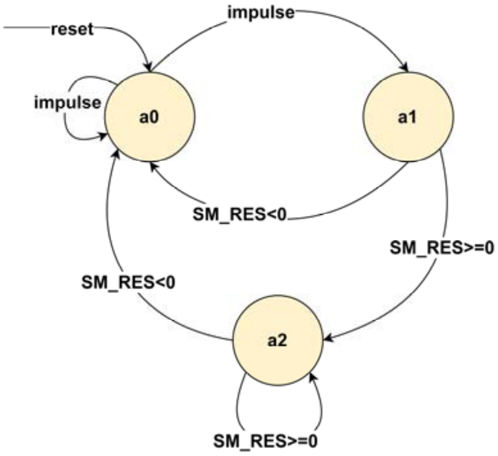

Figure 3 – State diagram of the computer control unit

An automaton template is one of the ways to describe a FSM in the hardware description language VHDL, in which the structure of the HDL model of the control unit of an online-computer is built on the basis of a state diagram.

The created HDL model of the control unit is an HDL template for the implementation of HDL models of online-computers. The control unit of the device is described by a state diagram that has three states a0, a1, a2.

At the signal reset = 1, the unit goes to state a0 and remains in it until the moment when the signal "impulse" appears.

After that, the unit goes to state a1. In state a1, pipelined computations are performed in the architecture's forward communication device components Block1.

If the content of the result adder SM_RES ≥ 0 becomes non-negative, then the device output bit appears at its output and the unit goes to state a2. If SM_RES ≥ 0 unit goes to a0 state.

In a2 pipeline calculations are performed in the feedback device components of the Block2 architecture.

If the value is SM_RES < 0, the unit goes to state a0. Also, in a2, the unit generates a signal to form the output bit of the device. The unit is in a2 if the value is SM_RES ≥ 0.

Figures 4, 5 present algorithm flowcharts of the operating unit.

Fig. 4 shows the basic algorithm flowchart of the operating unit for the development of state diagrams of specific power and root extraction functions' implementation.

Fig. 5 shows the algorithm flowchart of the operating unit for the implementation of the power function $y=[x^{\frac{3}{2}}+0.5]$.

The basic state diagram contains operator vertices a1 and a2 with computational microoperations in the device components.

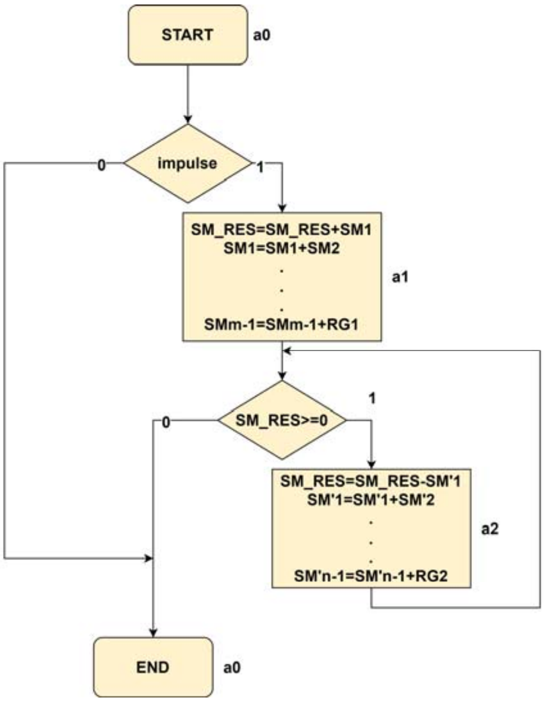

Figure 4 – Basic algorithm flowchart of the operating unit for implementation of a power function with an arbitrary fractional exponent

Figure 6 shows the algorithm flowchart of the operating unit for the implementation of the square root function.

Let's consider a block diagram of the online-computer.

As the result of the study, a block diagram of an online-computer was developed for hardware implementation, which includes two blocks: an impulse detector and a block of a bit-stream online-computer (Fig. 7).

The impulse detector unit detects the bits of the in-put stream x and sets the impulse = 1 signal at the output, which will be received by the arithmetic unit of the computer for further processing. The arithmetic block of the online-computer performs operations of raising the argument x to the power of a fraction-al-rational exponent and issues Ready signal when the block is ready to accept the next bit for processing. The result of the unit's operation is the output bit stream y, which is the result of the power function calculation.

DOI 10.15588/1607-3274-2024-1-6

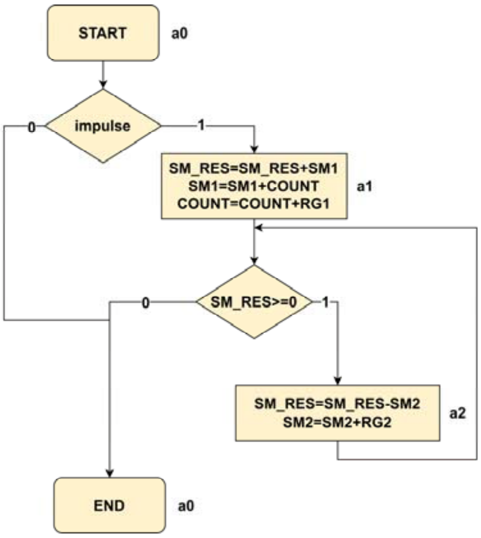

Figure 5 – Algorithm flowchart of the operating unit for implementation of power function $y=[x^{\frac{3}{2}}+0.5]$

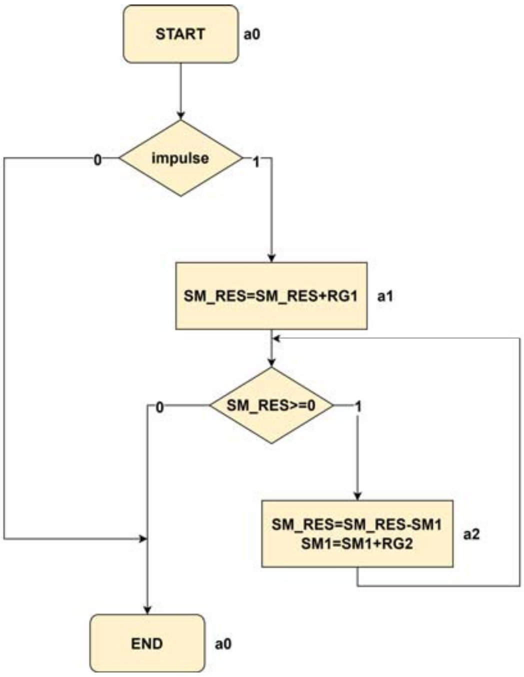

Figure 6 – Algorithm flowchart of the operating unit for square root function implementation

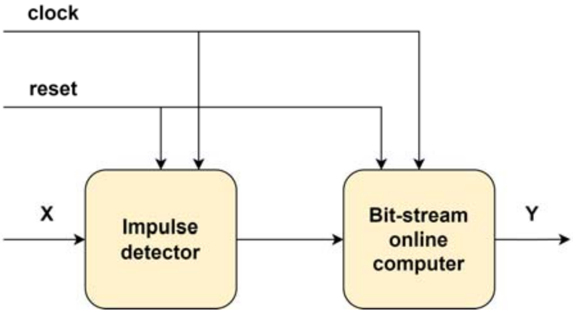

Figure 7 – Online-computer block diagram

4 EXPERIMENTS

The approbation was carried out through hardware implementation and evaluation of the created design models using the example of a bit-stream real-time computers for power functions and the square root function reproduction.

Automata models of the devices were built in the hardware description language VHDL, the device was synthesized and implemented in the Xilinx FPGA, hardware costs were estimated, and modelling was carried out using Active-HDL.

Let's consider an experimental study of online-computer of power functions.

The online-computer of power functions can function in two modes.

Mode 1. Operation of the online-computer in the mode of sampling of bits from the bit stream data, i. e. number divider mode when exponents $m<n$.

Mode 2. Operation of the online-computer in the bit sequences series generation mode when exponents $m>n$.

An experimental study of the online power functions computer in the mode of sampling of bits from the bit stream data is given in [23].

For the approximating function $y=[x^{\frac{2}{3}}+0.5]$ sample values x_y was determined from the input bit stream using the formula $x_y=[(y-0.5)^{\frac{3}{2}}]+1$. When substituting the y=1,2,3,4,5 values, sample values are obtained x_y = 1,2,4,7,10 respectively.

It is noted that the inequality $2^3x_y^2\geq(2y-1)^3$ is implemented in the online power function computer. The calculation of the left and right parts of the inequality is performed on the basis of the algorithm of pipeline calculations, which involves the calculation of arithmetic series of the second and third orders and of their arithmetic differences, respectively. At the same time, device synthesis is carried out by reducing the order of differences.

The study of online-computer of power functions in the bit sequences series generation mode was conducted.

For approximating function $y=[x^{\frac{3}{2}}+0.5]$, using the formula $x_y=[(y-0.5)^{\frac{2}{3}}]+1$ the number of bits in the series of bit sequences that are formed at the output of the device when each bit of x is applied to its input was determined.

If $y=1$, then $x=1$, that is, the first bit of the input sequence x will be selected and sent to the output of the device. If $y=2$ and $y=3$, then $x=2$, which means the formation of two series of bit sequences of 2 bits each. If $y=4$ and $y=5$, then $x=3$, that is, two series of bit sequences of three bits will be formed, and so on.

The inequality realized in device $2^2x_y^3\geq(2y-1)^2$. In order to calculate the arithmetic series of the second and third and orders values $x=\overline{0.8}$, $x=\overline{1.7}$ should substi-

DOI 10.15588/1607-3274-2024-1-6

tuted to the left and right parts of the considered inequality, respectively.

For the function $2^2 x_y^3$ the arithmetic series of the third order and the arithmetic series of differences of the first, second and third orders have the form:

f: 0, 4, 32, 108, 256, 500, 864, 1372, …;
Δ: 4, 28, 76, 148, 244, 364, 508, …;
Δ^2: 24, 48, 72, 96, 120, 144, ….;
Δ^3: 24, 24, 24, 24, 24, …

Considering function $(2y–1)^2$ the arithmetic series of the second order and the arithmetic series of first and second order differences have the form:

f: 1, 9, 25, 49, 81, 121, 169, …;
Δ: 8, 16, 24, 32, 40, 48, …;
Δ^2: 8, 8, 8, 8, 8, …

The results of calculating the function values for first seven input bits are presented in Table 1.

The device architecture includes components: adders SM_RES, SM1; counter Count; register RG1 which implement function $2^2 x_y^3$ and adders SM_RES, SM2; register RG2 which implement function $(2y \ -1)^2$.

Table 1 – Calculation of power function

X	Value of function Y
X=1	$y=[1^{\frac{3}{2}}+0.5]=[1.5]=1$
X=2	$y=[2^{\frac{3}{2}}+0.5]=[3.33]=3$
X=3	$y=[3^{\frac{3}{2}}+0.5]=[5.7]=5$
X=4	$y=[4^{\frac{3}{2}}+0.5]=[8.5]=8$
X=5	$y=[5^{\frac{3}{2}}+0.5]=[11.68]=11$
X=6	$y=[3^{\frac{3}{2}}+0.5]=[5.7]=5$
X=7	$y=[7^{\frac{3}{2}}+0.5]=[19.02]=19$

Initialization of architecture components: Count=24, SM1=4, SM_RES=—1, SM2=8, RG1=24, RG2 = 8.

Table 2 shows the results of computing process in components of the device.

The results of the calculation process in the components coincide with the results of the calculation of the power function for the first 7 bits of bit-stream data.

Let's consider an experimental research and approbation results by hardware implementation of online-computer of square root function.

For approximating function $y=[\sqrt{x}+0.5]$ sample values x_y were determined, which are selected from the input bit stream and fed to the input of the device, by the formula $x_y=[(y-0.5)^2]+1$.

Table 2 – Computing process in device components

X	SM_RES	Y	SM_1	Count	SM_2
1	–1 + 4 =3 3– 8 = –5	1	4 + 24 = 28	24 + 24 = 48	8 + 8 = 16
2	–5 + 28 = 23 23 –16 = 7 7 – 24 =–17	1 1	28 + 48 = 76	48 + 24 = 72	16 + 8 = 24 24 + 8 = 32
3	–17 + 76 = 59 59 – 32 = 27 27 – 40 = –13	1 1	76 + 72 = 148	72 + 24 = 96	32 + 8 = 40 40 + 8 = 48
4	–13 + 148 = 135 135 – 48 = 87 87 – 56 = 31 31 – 64 = –33	1 1 1	148 + 96 = 244	96 + 24 = 120	48 + 8 = 56 56 + 8 = 64 64 + 8 = 72
5	–33 + 244 = 211 211 – 72 = 139 139 – 80 = 59 59 – 88 = –29	1 1 1	244 + 120 = 364	120 + 24 = 144	72+8 = 80 80+8 = 88 88+8 = 96
6	–29 + 364 = 335 335 – 96 = 239 239 – 104 = 135 135 – 112 = 23 23 – 120 = – 97	1 1 1 1	364 + 144 = 508	144 + 24 = 168	96 + 8 = 104 104 + 8 = 112 112 + 8 = 120 120 + 8 = 128
7	–97 + 508 = 411 411 – 128 = 283 283 – 136 = 147 147 – 144 = 3 3 – 152 = – 149	1 1 1 1	508 + 168 = 676	168 + 24 = 192	128 + 8 = 136 136 + 8 = 144 144 + 8 = 152 152 + 8 = 160

DOI 10.15588/1607-3274-2024-1-6

When substituting the values $y = 1, 2, 3, 4$ the sample values x_y = 1, 3, 7, 13, are obtained, which will be selected from the input bit stream and applied to the output of the device. In the development of the pipeline architecture of the root extraction function computer, the algorithm of pipeline calculations is used, which is based on finding arithmetic series and their differences. The inequality $2^2 x_y \geq (2y_k - 1)^2$ is realized in the device.

When substituting y = 1, 2, 3, ... into the right-hand side of the inequality, the value of the function is obtained, which is an arithmetic series of the second order and the corresponding values of the difference series of the first and second orders:

f: 1, 9, 25, 49, 81, 121, 169, 225, 289, 361, …;
Δ: 8, 16, 24, 32, 40, 48, 56, 64, 72, …;
Δ^2: 8, 8, 8, 8, 8, 8, 8, 8, …

The results of calculating the function values for first 13 input bits are presented in Table 3.

Table 3 – Calculation of square root function

X	Value of function Y
X=1	$y = [\sqrt{1} + 0.5] = \left[1.5\right] = 1$
X=2	$y = [\sqrt{2} + 0.5] = \left[1.91\right] = 1$
X=3	$y = [\sqrt{3} + 0{,}5] = \left[2.23\right] = 2$
X=4	$y = [\sqrt{4} + 0.5] = \left[2.5\right] = 2$
X=5	$y = [\sqrt{5} + 0.5] = \left[2.74\right] = 2$
X=6	$y = [\sqrt{6} + 0.5] = \left[2.95\right] = 2$
X=7	$y = [\sqrt{7} + 0{,}5] = \left[3.15\right] = 3$
X=8	$y = [\sqrt{8} + 0.5] = \left[3.33\right] = 3$
X=9	$y = [\sqrt{9} + 0.5] = \left[3.5\right] = 3$
X=10	$y = [\sqrt{10} + 0.5] = \left[3.66\right] = 3$
X=11	$y = [\sqrt{11} + 0.5] = \left[3.82\right] = 3$
X=12	$y = [\sqrt{12} + 0.5] = \left[3.96\right] = 3$
X=13	$y = [\sqrt{13} + 0.5] = \left[4.1\right] = 4$

Table 4 shows the results of computing process in online-computer components.

Table 4 – Computing process in the device components

X	SM_RES	Y	SM_1
1	–1 + 4 = 3 3 – 8 = –5	1	8 + 8 = 16
2	–5 + 4 = –1		
3	–1 + 4 = 3 3 – 16 = –13	1	16 + 8 = 24
4	–13 + 4 = –9		
5	–9 +4 = –5		
6	–5 + 4 = –1		
7	–1 + 4 = 3 3 – 24 = –21	1	24 + 8 = 32
8	–21 +4 = –17		
9	–17 + 4 = –13		
10	–13 + 4 = –9		
11	–9 + 4 = –5		
12	–5 + 4 = –1		
13	–1 + 4 = 3 3 – 32 = –29	1	32 + 8 = 40

5 RESULTS

Verification of the models of the studied online-computers was performed using the Active-HDL modelling system. Behavioral models of online computers were obtained based on the results of theoretical calculations in Section 4.

Fig. 8 shows the results of modelling the behavioural model of the device in the mode of sampling of bits from the bit stream data, i. e. number divider mode. Presented waveform shows, that when 7 bits of the input stream x are supplied to the input of the computer, 4 bits of the output bit stream y are generated at the output of the device, and therefore 1, 2, 4 and 7 bits corresponding to the sample numbers x_y.

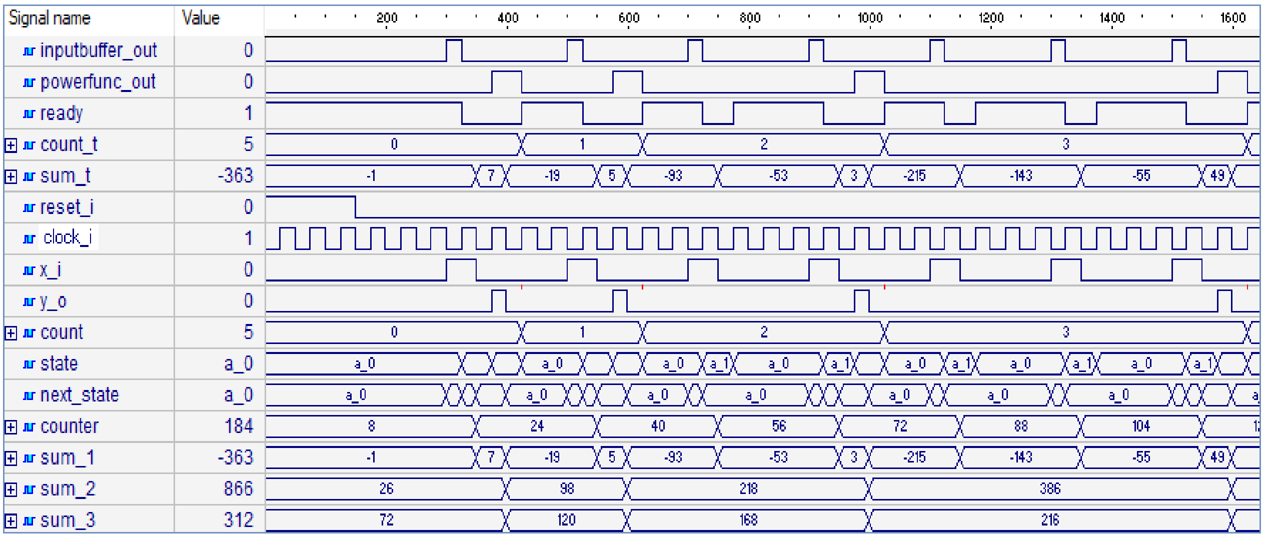

Figure 8 – The behavioral model of a power function online-computer in the mode of sampling of bits from the bit stream data

© Larchenko L. V., Parkhomenko A. V., Larchenko B. D., Korniienko V. R., 2024
DOI 10.15588/1607-3274-2024-1-6

Fig. 9 shows the results of modelling the behavioural model of the bit-stream online-computer of the power function in the mode of generating bit sequences series, which coincided with the results of theoretical calculations.

The detailed waveform of the behavioural model of the online-computer contains information signals and register values of the device components that coincide with the results of the computational process.

At the output of the computer, a series of pulse sequences are generated for each input bit *x* starting from the second bit, which coincides with Table 2. When the second and third bits are applied to the input of the device, two series of two bits each are generated at the output. When the forth and fifth bits are applied to the input, two series of bits of three bits each are generated at the output of the device, and so on.

Waveform confirms the correct operation of the online-computer in the mode of generating series of bit sequences.

Fig. 10 shows the detailed waveform of the behavioural model of the square root function online-computer.

The waveform demonstrates the process of supplying 13 bits of the input stream x to the input of the computer, from which 1, 3, 7, 13 bits were selected and sent to the output of the device in the form of the output stream y (*y*=4), which is the result of playing the square root function. The values of the registers coincide with the results of the computing process in the device components in accordance with Table 4.

The synthesis of the devices was carried out using CAD XILINX ISE. The schematic implementation of the computer is carried out on the XC3S100E series Xilinx SPARTAN 3E FPGA, which utilized approximately 6% of platform's resources. RTL–scheme of the synthesized power function computer is shown in Fig. 11. The maximum frequency of the devices is 125 MHz.

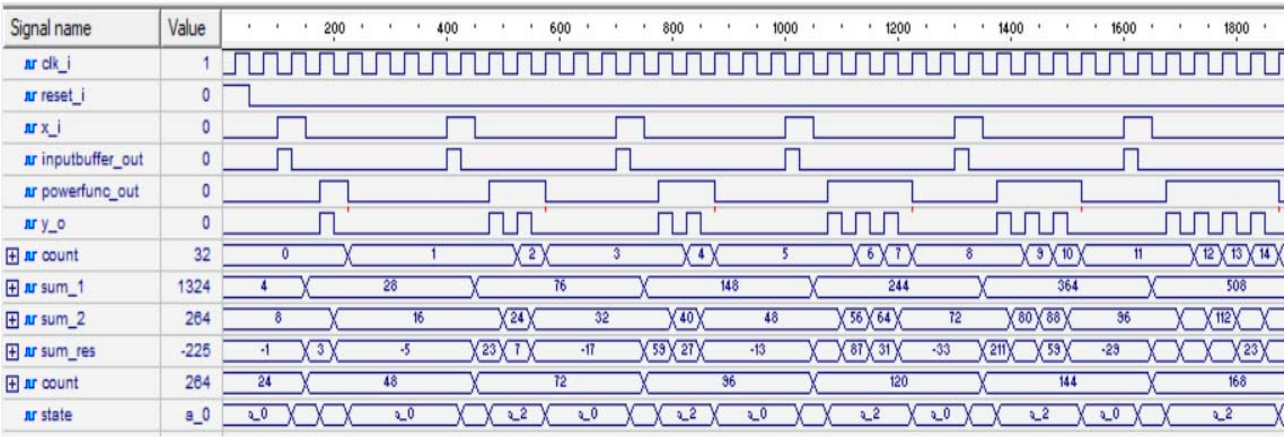

Figure 9 – The behavioral model of the bit-stream online-computer of the power function in the mode of generating bit sequences series

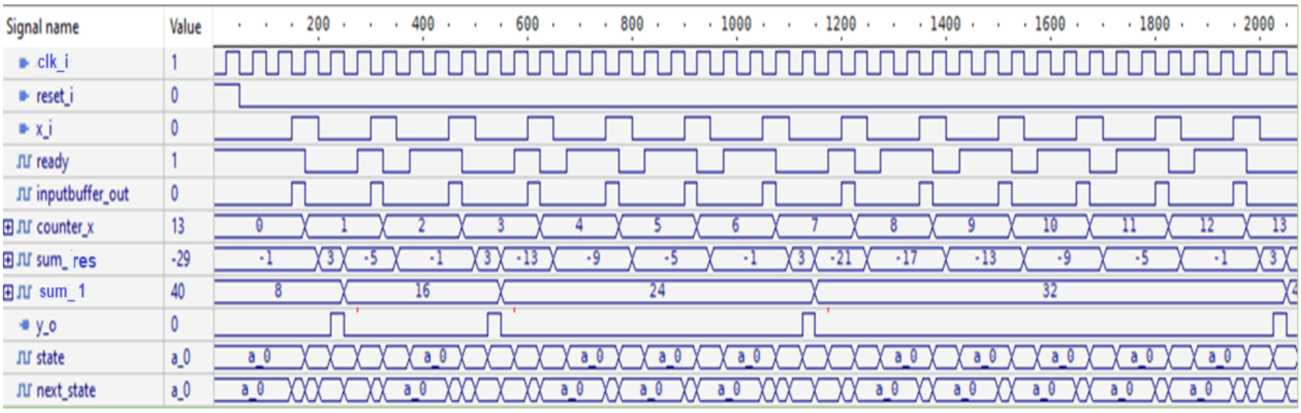

Figure 10 – The behavioural model of the bit-stream online-computer of root extraction function in the mode of sampling of bits from the bit stream data

DOI 10.15588/1607-3274-2024-1-6

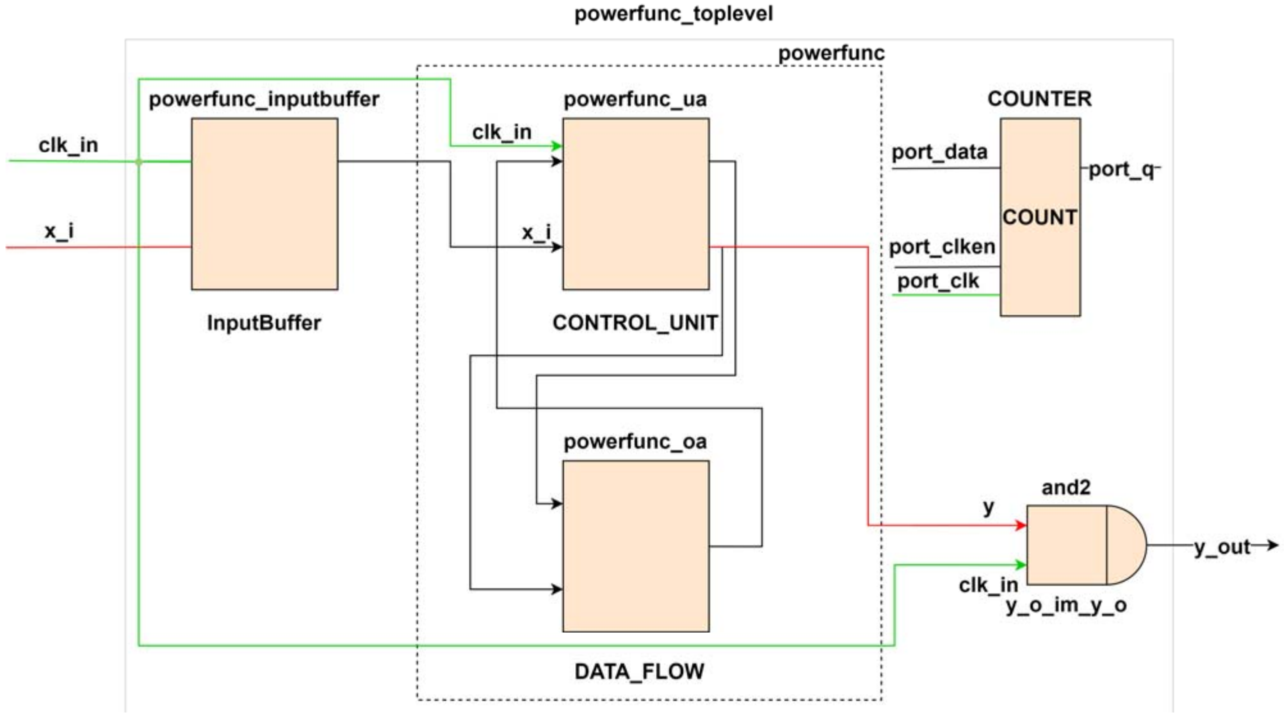

Figure 11 – RTL–scheme of the synthesized power function computer

6 DISCUSSION

The conducted research solves the actual scientific and practical problem of developing methods and models of designing hardware computers of mathematical functions of a specific class with bit streams data which perform calculations in real-time. In online-computers, a streaming method of calculations is organized with parallel-sequential execution of conversions over single bits of the input stream in accordance with a given function.

Increasing the accuracy of calculations and the speed of obtaining results in bit-stream online-computers of power functions and root extraction function is due to the development of improved mathematical models of computers based on the method of forming increments of ascending step functions with the minimization of absolute errors.

Expanding the functionality of online-computers is achieved by the construction of reconfigurable pipeline architectures based on the proposed generalized architecture of the computer, which allowed to develop a unified approach to their automated synthesis using hardware description languages.

Increasing the clarity and invariance of the implementation in formal languages of programming and hardware description is achieved due to the clarity and correctness of the algorithms for the implementation of functions by the proposed graph models of online-computers based on FSM.

CONCLUSIONS

As a result of the work, the generalized architecture of an online-computer was proposed, algorithm flowchart implementations of specific functions and the state diagram of Moore model control automaton of the computer's arithmetic unit were created.

Automata HDL models in the form of automata patterns were developed. The performance of the developed models of online-computers was confirmed by checking the results using the verification of behavioural models using Active-HDL CAD, automated synthesis and implementation in the Xilinx FPGA platform.

The considered functional online-computers are effective from the point of view of calculation accuracy, simplicity of technical implementation, and universality of the architecture.

The scientific novelty of the obtained results lies in the fact that the improved mathematical models of bit-stream online-computers of power functions and root extraction function were developed using the analysis method of forming increments of ascending step functions based on inverse functions with minimization of calculation errors. Automata models of bit-stream online-computers of elementary mathematical functions, characterized by graph models, were proposed, which made it possible to ensure the clarity and consistency of function implementation algorithms.

The practical significance of the obtained results lies in the fact that the use of the developed mathematical, architectural and automata design models of bit-stream online-computers of elementary mathematical functions ensures an increase in the accuracy of the calculation of mathematical functions, reduces equipment costs and development time, which ultimately increases the efficiency of the design process. It also consists in the development of automata models of bit-stream functional computers, which are formed on the basis of the finite state machine of the Moore model, which made it possible to create the

DOI 10.15588/1607-3274-2024-1-6

same type of graph models and HDL models in the form of an automata pattern for the implementation of algorithms of reproducible functions of online-computers with CAD tools.

The prospects for the further research involve the application of the developed design models to other signal encoding methods. Also, the mentioned approaches might perfectly suit to be deployed on the SoCs such as Xilinx ZYNQ which contains Programming System (ARM) part and Programmable Logic (PL) component with the optimal computation distribution between them. This includes the further research directions of using partial reconfiguration for deploying various bit-stream specific hardware accelerators with the capabilities of resource management on ARM side.

ACKNOWLEDGEMENTS

The work was carried out within the framework of the state budget research topic of the Kharkiv National University of Radio Electronics "Smart cyber university – Cloud-Mobile services for managing scientific and educational processes l" (state registration 0017U002524, order of the Ministry of Education of Ukraine) on the basis of the Design automation department, KNURE.

And out within the framework of the department research topic of the National University Zaporizhzhia Polytechnic "Methods, models and software tools of intelligent computing" (state registration number 0121U112837, order of the Ministry of Education of Ukraine) on the basis of the of Software Tools department, NUZP.

REFERENCES

1. Korniienko V., Larchenko L., Parkhomenko A., Larchenko B. Design Models of Bit-Stream Online-Computers of Elementary Mathematical Functions, *12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Sep 2023*. Dortmund, Germany, pp. 585–589. DOI : 10.1109/IDAACS58523.2023.10348885.
2. Al-Makhles D., Patel N., Swain A. Bitstream Control System: Stability and Experimental Application, *2013 International Conference on Applied Electronics, Sep 2013*. Pilsen, Czech Republic, pp. 19–24. ISSN: 1803-7232.
3. Al-Makhles D., Patel N., Swain A. Conventional and Hybrid Bit-Stream in Real-Time System, *11th Workshop on Intelligent Solutions in Embedded Systems (WISES), Sep 2013*. Pilsen, Czech Republic, pp. 1–6. ISBN:978-3-00-042899-9.
4. Li P., Lilja D. J., Qian W., Riedel M. D. Bazargan. K. Logical Computation on Stochastic Bit Streams with Linear Finite-State Machines, *IEEE Transactions on Computers*, 2014, Volume 63(6), pp. 1474–1486. DOI : 10.1109/TC.2012.231.
5. Yan M., Nan X., Liru H., Jinghua T., Weimin C. Development on Intelligent Small-Flow Target Flow Meter, *2009 International Forum on Computer Science-Technology and Applications, Dec 2009*. Chongqing, China, pp. 315–317. DOI : 10.1109/IFCSTA.2009.317.
6. Stanley M., Gervais-Ducouret S., Adams J. T. Intelligent Sensor Hub Benefits for Wireless Sensor Networks, *2012 IEEE Sensors Applications Symposium, Feb 2012*. Brescia, Italy, pp. 1–6. DOI : 10.1109/SAS.2012.6166299.
7. Gulin A. I., Safyannikov N. M., Bureneva O. I., Kaydanovich A. Yu. Assurance of Fault-Tolerance in Bit-Stream Computing Converters, 16th IEEE East-West Design & Test Symposium, Sep 2018. Kazan, Russia, pp. 418–421. DOI : 10.1109/EWDTS.2018.8524812.
8. Ueno K., Asai T., Amemiya Y. Temperature-to-Frequency Converter Consisting of Subthreshold Mosfet Circuits for Smart Temperature-Sensor LSIs, *2009 TRANSDUCERS International Solid-State Sensors, Jun 2009, Actuators and Microsystems Conference*. Denver, CO, USA, pp. 2433–2436. DOI : 10.1109/SENSOR.2009.5285436.
9. Arbet D., Kováč M., Stopjaková V., Potočný M. Voltage-to-Frequency Converter for Ultra-Low-Voltage Applications, *42nd International Convention on Information and Communication Technology, Electronics and Micro-electronics (MIPRO), May 2019*. Opatija, Croatia, pp. 53–58. DOI : 10.23919/MIPRO.2019.8756910.
10. Areekath L., George B., Reverter F. An Auto-Balancing Capacitance-to-Pulse-Width Converter for Capacitive Sensors, *IEEE Sensors Journal*, 2021, Volume 21(1), pp. 765–775. DOI : 10.1109/JSEN.2020.3014206.
11. Safyannikov N., Bureneva O., Bit-Stream Functional Converters for Decentralized Sensor Systems, *9th Mediterranean Conference on Embedded Computing (MECO), Jun 2020*. Budva, Montenegro, pp. 1–4. DOI : 10.1109/MECO49872.2020.9134176.
12. Bureneva O., Kupriyanov M., Safyannikov N. Bit Streaming Processing Algorithms for Intelligent Hardware Converters, *Applied Sciences*, 2021, Volume 186, pp. 194–201. DOI : 10.1016/j.procs.2021.04.138.
13. Bureneva O., Mironov S., Safyannikov N. Bit-Stream Approximating Devices Based on Counters with Weight Inputs, *11th Mediterranean Conference on Embedded Computing (MECO), Jun 2022*. Budva, Montenegro, pp. 1–4. DOI : 10.1109/MECO55406.2022.9797145.
14. Bureneva O. I. Stream Tracking Devices for Soft Measurements Implementation, *XX IEEE International Conference on Soft Computing and Measurements (SCM), May 2017*. St. Petersburg, Russia, pp. 614–616. DOI : 10.1109/SCM.2017.7970666.
15. Chelebaev S. V., Chelebaeva Y. A. Converters Structures Synthesis of Time-and-Frequency Signals Parameters in the Code of Two Variables on the Radial Basis Network, *5th Mediterranean Conference on Embedded Computing (MECO), Jun 2016*. Budva, Montenegro, pp. 339–342. DOI : 10.1109/MECO.2016.7525776.
16. Aggarwal S, Meher P. K., Khare K. Concept, Design, and Implementation of Reconfigurable CORDIC, *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, 2016, Volume 24(4), pp. 1588–1592. DOI : 10.1109/TVLSI.2015.2445855.
17. Meher P. K., Valls J., Juang T. B., Sridharan K., Maharatna K. 50 Years of CORDIC: Algorithms, Architectures, and Applications, *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2009, Volume 56(9), pp. 1893–1907. DOI : 10.1109/TCSI.2009.2025803.
18. Kumar P. A. FPGA Implementation of the Trigonometric Functions Using the CORDIC Algorithm, *5th International Conference on Advanced Computing & Communication Systems (ICACCS), Mar 2019*. Coimbatore, India, pp. 894–900. DOI : 10.1109/ICACCS.2019.8728315.
19. Sapper A. N., Soares L., Costa E., Bampi S. Exploring the Combination of Number of Bits and Number of Iterations for a Power-Efficient Fixed-Point CORDIC Implementation, *24th IEEE International Conference on Electronics, Circuits*

DOI 10.15588/1607-3274-2024-1-6

and Systems (ICECS), Dec 2017. Batumi, Georgia, pp. 302–305. DOI : 10.1109/ICECS.2017.8292079.

20. Xue Y., Ma Z. Design and Implementation of an Efficient Modified CORDIC Algorithm, *4th International Conference on Signal and Image Processing (ICSIP), Jul 2019.* Wuxi, China, pp. 480–484. DOI : 10.1109/SIPROCESS.2019.8868732.
21. Rodriguez-Garcia A., Pizano-Escalante L., Parra-Michel R., Longoria-Gandara O., Cortez J. Fast Fixed-Point Divider Based on Newton-Raphson Method and Piecewise Polynomial Approximation, *2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig), Dec 2013.* Cancun, Mexico, pp. 1–6. DOI : 10.1109/ReConFig.2013.6732291.
22. Dang Pham K., Horta E., Koch D. BITMAN: A Tool and API for FPGA Bitstream Manipulations, *2017 Design, Automation & Test in Europe Conference & Exhibition (DATE), Mar 2017.* Lausanne, Switzerland, pp. 894–897. DOI : 10.23919/DATE.2017.7927114.
23. Shkil A. S., Larchenko L. V., Larchenko B. D. Bit-Stream Power Function Online Computer, *2020 East-West Design & Test Symposium (EWDTS), Sep 2020.* Varna, Bulgaria, pp. 1–6. DOI : 10.1109/EWDTS50664.2020. 9224764.

Received 04.01.2024.
Accepted 28.02.2024.

УДК 004.31

МОДЕЛІ ПРОЄКТУВАННЯ БІТ-ПОТОКОВИХ ОНЛАЙН-ОБЧИСЛЮВАЧІВ ДЛЯ СЕНСОРНИХ КОМПОНЕНТІВ

Ларченко Л. В. – канд. техн. наук, доцент, доцент кафедри автоматизації проєктування обчислювальної техніки Харківського національного університету радіоелектроніки, Харків, Україна.

Пархоменко А. В. – канд. техн. наук, доцент, доцент кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

Ларченко Б. Д. – канд. техн. наук, асистент кафедри автоматизації проєктування обчислювальної техніки Харківського національного університету радіоелектроніки, Харків, Україна.

Корнієнко В. Р. – аспірант кафедри автоматизації проєктування обчислювальної техніки Харківського національного університету радіоелектроніки, Харків, Україна.

АНОТАЦІЯ

Актуальність. В даний час розподілені системи управління реального часу потребують створення пристроїв, які виконують онлайн-обчислювальні операції в оточенні датчика. Запропоновані online-обчислювачі елементарних математичних функцій можуть бути використані як компоненти для функціонального перетворення сигналів у вигляді імпульсних потоків, що надходять від вимірювальних датчиків з частотним виходом.

Мета роботи. Розробка математичних, архітектурних та автоматних моделей проєктування біт-потокових онлайн- обчислювачів елементарних математичних функцій з метою створення єдиного підходу до їх проєктування, завдяки якому можливо підвищити точність обчислення функцій, розширити функціональні можливості, зменшити апаратупні витрати та підвищити ефективність проєктування.

Метод. Розроблено математичні моделі пристроїв з використанням методу формування приростів ступінчастих функцій на основі обернених функцій з мінімізацією похибки обчислень. Розроблено автоматні моделі онлайн- обчислювачів на основі кінцевого автомата Мура, графові моделі яких дозволили забезпечити чіткість алгоритмів реалізації функцій, підвищити наочність та інваріантність реалізації на формальних мовах програмування та опису апаратури.

Результати. У статті наведено результати дослідження, розробки та практичної апробації моделей проєктування біт-потокових онлайн- обчислювачів степеневої функції та функції вилучення кореня. Запропоновано узагальнену архітектуру онлайн- обчислювача.

Висновки. Розглянуті функціональні онлайн- обчислювачі ефективні з точки зору точності обчислень, простоти технічної реалізації та універсальності архітектури.

КЛЮЧОВІ СЛОВА: функціональне перетворення, бітові дані, бітові обчислення, математична модель, кінцевий автомат, FPGA, SoC.

ЛІТЕРАТУРА

1. Design Models of Bit-Stream Online-Computers of Elementary Mathematical Functions / [V. Korniienko, L. Larchenko, A. Parkhomenko et al.] // 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems, Technology and Applications (IDAACS), Sep 2023, Dortmund, Germany. – Germany. – P. 585–589. DOI : 10.1109/IDAACS58523.2023.10348885.
2. Al-Makhles D. Bitstream Control System: Stability and Experimental Application / D. Al-Makhles, N. Patel, A Swain // 2013 International Conference on Applied Electronics, Sep 2013, Pilsen, Czech Republic. – Czech Republic. – P. 19–24. ISSN: 1803-7232.
3. Al-Makhles, D. Conventional and Hybrid Bit-Stream in Real-Time System / D. Al-Makhles, N. Patel, A. Swain // 11th Workshop on Intelligent Solutions in Embedded Systems (WISES), Sep 2013, Pilsen, Czech Republic. – Czech Republic. – P. 1–6. ISBN:978-3-00-042899-9.
4. Logical Computation on Stochastic Bit Streams with Linear Finite-State Machines / [P. Li, D. J. Lilja, W. Qian et al.] // IEEE Transactions on Computers. – 2014. – Volume 63(6). – P. 1474–1486. DOI: 10.1109/TC.2012.231.
5. Development on Intelligent Small-Flow Target Flow Meter / [M. Yan, X. Nan, H. Liru et al.] // 2009 International Forum on Computer Science-Technology and Applications, Dec 2009, Chongqing, China. – China. – P. 315–317. DOI : 10.1109/IFCSTA.2009.317.
6. Stanley M. Intelligent Sensor Hub Benefits for Wireless Sensor Networks / M. Stanley, S. Gervais-Ducouret, J. T Adams // 2012 IEEE Sensors Applications Symposium, Feb 2012, Brescia, Italy. – Italy. – P. 1–6. DOI: 10.1109/SAS.2012.6166299.

DOI 10.15588/1607-3274-2024-1-6

7. Assurance of Fault-Tolerance in Bit-Stream Computing Converters / [A. I. Gulin, N. M. Safyannikov, O.I. Bureneva et al.] // 16th IEEE East-West Design & Test Symposium, Sep 2018, Kazan, Russia. – Russia. – P. 418–421. DOI: 10.1109/EWDTS.2018.8524812.
8. Ueno K. Temperature-to-Frequency Converter Consisting of Subthreshold Mosfet Circuits for Smart Temperature-Sensor LSIs / K. Ueno, T. Asai, Y. Amemiya // 2009 TRANSDUCERS International Solid-State Sensors, Actuators and Microsystems Conference, Jun 2009, Denver, CO, USA. – USA. – P. 2433–2436. DOI: 10.1109/SENSOR.2009.5285436.
9. Voltage-to-Frequency Converter for Ultra-Low-Voltage Applications / [D. Arbet, M. Kováč; V. Stopjaková et al.] // 42nd International Convention on Information and Communication Technology, Electronics and Micro-electronics (MIPRO), May 2019, Opatija, Croatia. – Croatia. – P. 53–58. DOI : 10.23919/MIPRO.2019.8756910.
10. Areekath L. An Auto-Balancing Capacitance-to-Pulse-Width Converter for Capacitive Sensors / L. Areekath, B. George, F. Reverter // IEEE Sensors Journal. – 2021. – Volume 21(1). – P. 765–775. DOI: 10.1109/JSEN.2020.3014206.
11. Safyannikov, N. Bit-Stream Functional Converters for Decentralized Sensor Systems / N. Safyannikov, O. Bureneva // 9th Mediterranean Conference on Embedded Computing (MECO), Jun 2020, Budva, Montenegro. – Montenegro. – P. 1–4. DOI: 10.1109/MECO49872.2020.9134176.
12. Bureneva O. Bit Streaming Processing Algorithms for Intelligent Hardware Converters / [O. Bureneva, M. Kupriyanov, N. Safyannikov] // Applied Sciences. – 2021. – Volume 186. – P. 194–201. DOI: 10.1016/j.procs.2021.04.138.
13. Bureneva O. Bit-Stream Approximating Devices Based on Counters with Weight Inputs / O. Bureneva, S. Mironov, N. Safyannikov // 11th Mediterranean Conference on Embedded Computing (MECO), Jun 2022, Budva, Montenegro. – Montenegro. – P. 1–4. DOI: 10.1109/MECO55406.2022.9797145.
14. Bureneva O. I. Stream Tracking Devices for Soft Measurements Implementation / O. I. Bureneva // XX IEEE International Conference on Soft Computing and Measurements (SCM), May 2017, St. Petersburg, Russia. – Russia. – P. 614–616. DOI : 10.1109/SCM.2017.7970666.
15. Chelebaev S. V. Converters Structures Synthesis of Time-and-Frequency Signals Parameters in the Code of Two Variables on the Radial Basis Network / S. V. Chelebaev, Y. A. Chelebaeva // 5th Mediterranean Conference on Embedded Computing (MECO), Jun 2016, Budva, Montenegro. – Montenegro. – P. 339–342. DOI: 10.1109/MECO.2016.7525776.
16. Aggarwal, S. Concept, Design, and Implementation of Reconfigurable CORDIC / S. Aggarwal, P. K. Meher, K. Khare // IEEE Transactions on Very Large-Scale Integration (VLSI) Systems. – 2016. – Volume 24(4). – P. 1588–1592. DOI: 10.1109/TVLSI.2015.2445855.
17. 50 Years of CORDIC: Algorithms, Architectures, and Applications / [P. K. Meher, J. Valls, T. B. Juang et al.] // IEEE Transactions on Circuits and Systems I: Regular Papers. – 2009. – Volume 56(9). – P. 1893–1907. DOI: 10.1109/TCSI.2009.2025803.
18. Kumar, P. A. FPGA Implementation of the Trigonometric Functions Using the CORDIC Algorithm / P. A. Kumar // 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Mar 2019, Coimbatore, India. – India. – P. 894–900. DOI: 10.1109/ICACCS.2019.8728315.
19. Exploring the Combination of Number of Bits and Number of Iterations for a Power-Efficient Fixed-Point CORDIC Implementation / [A. N. Sapper, L. Soares, E. Costa et al.] // 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Dec 2017, Batumi, Georgia. – Georgia. – P. 302–305. DOI: 10.1109/ICECS.2017.8292079.
20. Xue Y. Design and Implementation of an Efficient Modified CORDIC Algorithm / Y. Xue, Z. Ma // 4th International Conference on Signal and Image Processing (ICSIP), Jul 2019, Wuxi, China. – China. – P. 480–484. DOI: 10.1109/SIPROCESS.2019.8868732.
21. Fast Fixed-Point Divider Based on Newton-Raphson Method and Piecewise Polynomial Approximation / [A. Rodriguez-Garcia, L. Pizano-Escalante, R. Parra-Michel et al.] // 2019 International Conference on Reconfigurable Computing and FPGAs (ReConFig), Dec 2013, Cancun, Mexico. – Mexico. – P. 1–6. DOI: 10.1109/ReConFig.2013.6732291.
22. Dang Pham, K. BITMAN: A Tool and API for FPGA Bitstream Manipulations / K. Dang Pham, E. Horta, D. Koch // 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE), Mar 2017, Lausanne, Switzerland. – Switzerland. – P. 894–897. DOI: 10.23919/DATE.2017.7927114.
23. Shkil, A. S. Bit-Stream Power Function Online Computer / A. S. Shkil, L. V. Larchenko, B. D. Larchenko // 2020 IEEE East-West Design & Test Symposium (EWDTS), Sep 2020, Varna, Bulgaria. – Bulgaria. – P. 1–6. DOI : 10.1109/EWDTS50664.2020. 9224764.

DOI 10.15588/1607-3274-2024-1-6